DEP SEQUENCE CONTROL ROUTINE SPECIFICATION

1. SCOPE

This document describes the functional capabilities and operational characteristics of the MDI Sequence Control Routine (SCR).

2. INTRODUCTION

SCR is the flight segment software tool for sequencing of the MDI instrument through a set of configurations and associated Image Processor (IP) computations. SCR is one facet of the flight software contained in the MDI Dedicated Experiment Processor (DEP). SCR is a task specific automaton of relatively limited scope oriented toward setting the instrumentation in particular configurations and taking images with the camera. SCR contains 16 "general purpose" sequence registers (SR's) that are used for flow control (loops and branches). Register operations also include some simple arithmetic capabilities that are required to implement flow control. SCR does no verification of sequences prior to execution time, nor does it test for consistency in the utilization of SR's. It is the responsibility of the ground based segment to manage the use of sequence registers.

3. GENERAL DESCRIPTION

A DEP command starts SCR. The command contains a pointer to a list of SCR instructions called functions. Most functions simply manipulate internal parameters values that control flow and branching. These execute immediately. There are three special functions that interact with the instrumentation and control the timing of the sequence. These are wait for time mark (TIM), configure the instrument (CON), and take a picture (TAP). TIM waits for a time mark. Time marks are relative to the current reference time. The reference time is set by the SRT instruction to the nearest observing interval (typically one minute) that is greater than current Local On Board Time (LOBT). Time mark resolution is the DEP time increment, i.e. 1/16 second. CON configures the optics for the next TAP. SCR will not execute a CON until the instrument is ready to be reconfigured. Details of the CON function are given in ¶4.2.4. TAP initiates a sequence of DEP operations that cause a camera image to be taken and sent to the image processor.

Once during each DEP time increment the main software program invokes the SCR. SCR executes functions until it encounters any TAP or CON, or a TIM with a time that is greater than current time. In these cases, control returns to main.

SCR operates completely in the software domain, i.e., SCR does not access the MDI instrumentation directly. It interacts with other portions of the MDI flight software by accessing memory locations that hold desired and actual configurations of various electrical and mechanical devices.

4. FUNCTIONS

4.1. FUNCTION SUMMARY

Table 1 contains a summary SCR functions. The mnemonic codes are those used by the various components of the ground based support segment. Functions contain up to three operands. Operand types may be Register (R), Immediate (I), or Pointer (&). Registers operands are specified in a byte. Immediate operands are specified by the next two bytes in the sequence list. Pointer operands use the next two immediate bytes as a 16 bit address. Pointer addresses are relative to the sequence buffer in DEP memory.

4.2. ADDITIONAL DETAILS

Table 1 contains a sufficient definition of most functions. The sections that follow add required details.

4.2.1. REGISTERS

SCR contains 16 registers. Register functions and supporting telecommands allow registers to vary the execution flow of a sequence. This is normally predetermined, i.e., a set of functions execute a number of times then a branch of some sort occurs. Since registers can also be effected by telecommands, asynchronous flow control is possible. For example, a sequence may initially set a register to 0 and include a branch if the register value is non-zero. One can set the register to non-zero thus forcing the branch by sending the appropriate telecommand. The configure function allows registers as well as explicit pointer values; so, for example, one can change the IP program invoked from a sequence without reloading the sequence.

4.2.2. CALL AND RETURN FUNCTIONS

The function set includes a call capability (CLL). CLL transfers control to a specified address. The RET function transfer control back to the function following the CLL. The nesting level is 3, i.e. there can be up to 3 return addresses. The CLL function includes a register mask. This mask determines which registers are to be restored by the RET[1]. The Sequence Identifier at the CLL is restored with RET.

---

[1] If the compiler input to the linker does not specify a mask, the linker sets the mask to $FF00_{16}$, i.e. registers 8 - F are restored. By default, therefore, registers 8 - F are local and 0-7 are "global". This is conventional only established by the linker. The design concept is that one would never or rarely specify a mask at the compiler level.

NAME	OP1	OP2	OP 3	DESCRIPTION	CODING
BEQ	R,&	R	R,I	Branch to OP1 if OP2 == OP3	00000ac0
BNE	R,&	R	R,I	Branch to OP1 if OP2 != OP3	00010ac0
BGT	R,&	R	R,I	Branch to OP1 if OP2 > OP3	00100ac0
BLT	R,&	R	R,I	Branch to OP1 if OP2 < OP3	00110ac0
BLE	R,&	R	R,I	Branch to OP1 if OP2 <= OP3	01000ac0
BGE	R,&	R	R,I	Branch to OP1 if OP2 >= OP3	01010ac0
BZ	R,&	R	R	Branch to OP1 if OP2 == 0	01100000
BNZ	R,&	R	R	Branch to OP1 if OP2 != 0	011010a0
UBR	R,&			Branch to OP1	011100a0
CLL	R,&	I		Call to OP1	011110a0
DDC†	R,I			Direct Device Command (1 arg)	100000a0
DDC†	R,I	R,I		Direct Device Command (2 arg)	10001ab0
SET	R	R,I		OP1 $\leftarrow$ OP2	100100b0
ADD	R	R,I		OP1 $\leftarrow$ OP2 + OP1	100110b0
SUB	R	R,I		OP1 $\leftarrow$ OP2 - OP1	101000b0
CON†	&	&	R,I	Configure; DMA & APU Control	101010c0
CON†	&	&	&	Configure; DMA & APU/TLM Control	10110000
CON†	&	&		Configure; DMA Control only	10111000
EXP	R,I			Set Exposure Time	110010a0
SID	I			Set Sequence ID	11010010
GRT	R			Get Relative Time	11100010
GIC	R			Get Interval Counter (Since SIC)	11100100
INC	R			OP1 $\leftarrow$ OP1 + 1	11100110
DEC	R			OP1 $\leftarrow$ OP1 - 1	11101000
CLR	R			OP1 $\leftarrow$ 0	11101010
STS				Stop Sequence	11110000
SRT				Set Reference Time	11110010
SIC				Clear and Start Interval Counter	11110100
TAP				Take a Picture	11110110
TAD				Take a Dark Frame	11111000
RET				Return from CLL	11111010
TIM	I,R			Wait for Time Mark	111111a1

a: Operand 1 Modifier: 0 = Register; 1 = Pointer or Immediate
b: Operand 2 Modifier: 0 = Register; 1 = Pointer or Immediate
c: Operand 3 Modifier: 0 = Register; 1 = Pointer or Immediate
†: The optional codes for DDC and CON are created by the linker based on number and type of parameters in instruction passed by the compiler.

Sequence Control Functions
Table 1

4.2.3. DIRECT DEVICE COMMAND FUNCTION

The Direct Device Command function (DDC) is a copy of the DEP block telecommand 16_{16} . The content of DDC is shown in Table 2. The first DDC operand is a word that specifies function and device. The low byte of the operand contains device code in bits 0-3 and function in 4-7. For example, CAL1 Clockwise is DDC X17. Functions include both primitive set position and integrated device commands. Primitive command executes single transaction with the device at the next DEP time cycle, while an integrated command executes several transactions over a number time cycles. In Table 2, (I) in the function description column indicates and integrated function.

DEVICE		FUNCTION		OPERAND
NAME	CODE	DESCRIPTION	CODE	
MTM1	0	NOP	0	
		Clockwise	1	
		Reset	2	
		Counter-Clockwise	3	
		Set Position	4	Yes
		Move CW[2] (I)	5	Yes
		Move CCW[2] (I)	6	Yes
		Move Default[2] (I)	7	Yes
MTM2	1	Same as MTM1		
PAW	4	Same as MTM1		
CAL1	7	Same as MTM1		
CAL2	8	Same as MTM1		
DOOR	5	NOP	0	
		Open with Motor 1	1	
		Open with Motor 2	2	
		Close with Motor 1	3	
		Close with Motor 1	4	
		Open with Both	5	
		Close with Both	6	
		Reset	7	
SHUTTER	2	Full Disk (1X)	1	
		Reset	2	
		HI-Resolution (3X)	3	
		Load Low Byte	4	Yes
		Load High Byte	5	Yes
LEGS	6	NOP 1	0	
		Increment 1 (I)	1	Yes
		Decrement 1 (I)	2	Yes
		Reset 1	3	
		NOP 2	4	
		Increment 2 (I)	5	Yes
		Decrement 2 (I)	6	Yes
		Reset 2	7	

Direct Device Function Parameters
Table 2

[2]Move command is 2 primitives, i.e. Set Position followed by CW or CCW. The sequencer returns to the main program after the Set Position; so, these commands consume a DEP time interval.

4.2.4. CONFIGURE FUNCTION

The configure function contains pointers to blocks that define the configuration of the optics and the parameters to be sent to the IP via the camera header. There are three forms of CON. One format contains pointers to an Optics Configuration Block and a DMA control block. The second format also has the instruction queue address of an IP program. The third has a pointer to an APU/TM Control Block instead of a Queue address. The sections that follow define the format of these three blocks.

4.2.4.1. OPTICS CONFIGURATION BLOCK

The Optics Configuration Block (OCB) contains information about the devices moved most often during sequences, i.e. the MTM's and PAW. Tables 3-1 through 3-4 define the format on content of the OCB.

CONFIGURATION BITS
PAW POSITION
MTM0 POSITION
MTM1 POSITION

Optics Configuration Block
Table 3-1

BITS	7	6	5	4	3	2	1	0
	PAW DIR		MTM0 DIR		MTM1 DIR		SHUTTER	

Configuration Bits
Table 3-2

00	Default Direction
01	Clockwise
10	Counter-clockwise
11	No Motion

Direction Codes
Table 3-3

00	1X (Full Disk)
01	3X (High Resolution)
10	Default Type
11	Close (Dark Current)

Shutter Control
Table 3-4

4.2.4.1.1. MTM POSITIONS

MTM positions can be either absolute encoder addresses or lookup table indices. Values from 0 - 179 are treated as absolute, while values 255 - 211 are lookups. The lookup index is 255 - value. There is a separate lookup table for each MTM. Values 180 - 210 are treated the same as 0 - 30. The lookup tables are loaded by telecommand (See ¶5).

Each MTM has an offset. The final position sent to MTM is the requested address or lookup plus the offset modulo 180. The offsets are set by telecommand (See ¶5).

4.2.4.1.2. DEVICE ROTATION DIRECTIONS

Devices move in the direction indicated by the direction codes in the OCB. The default direction for a given device is the most recently specified direction for that device. Direction codes can also be set by telecommand (See ¶5).

4.2.4.2. DMA CONTROL BLOCK

The DMA control block contains information used by the Image Processor's CCD camera Interface to route the image data to IP memory. The DDC instruction contains a pointer to a block formatted as shown in tables 4-1 and 4-2. Further details on the IP camera interface can be found in MD330037.

7	6	5	4	3	2	1	0
DMA CMD				PAGE ADDRESS			
DMA CMD				PAGE ADDRESS			
DMA CMD				PAGE ADDRESS			
DMA CMD				PAGE ADDRESS			

DMA Control Block
Table 4-1

CMD BITS		FUNCTION
1	1	Subtract from Memory
1	0	Add to Memory
0	1	Write to Memory
0	0	No Operation

DMA CMD Bits
Table 4-2

4.2.4.3. APU/TM CONTROL BLOCK

One form of the CON function contains a pointer to an APU/TM control block. The format of the block is shown in Table 5. A detailed description of the parameters are presented in MD330037.

BITS	7	6	5	4	3	2	1	0
	N Parameter				K Parameter			
	PG ADR 2		PG ADR 1		1/2		TEL CMD	
						IFC CMD		
	APU CMD (QUEUE POINTER) (low byte)							
	APU CMD (QUEUE POINTER) (high byte)							

APU/TM Control Block
Table 5

4.2.5. TIMING CONTROL FUNCTIONS

Precise repetitive time cadence is the most important aspect of MDI sequence execution. The SCR design concept is based on idea of an observing interval and a reference time. During an observing interval a sequence takes a set of camera exposures in various instrument configurations. The pictures within the set are time tagged relative to the start of the observing interval. The same set is repeated for many observing intervals. Observing sequence consists of a number of different observing sets that execute for different periods of time. The observing interval at launch is one minute.

SCR contains several functions that control the cadence of sequence execution. In general, a sequence periodically sets a reference time using the SRT function, then controls the pace at which function execute with the TIM function. The unit of time within sequences is 1/16th second, i.e., the DEP execution interval.

4.2.5.1. SET REFERENCE TIME (SRT)

This function causes the reference time to be set to the next whole observing interval that is greater than the current LOBT[3]. Since the reference time is typically in the future, time marks can be negative.

4.2.5.2. WAIT FOR TIME MARK (TIM)

This function causes SRC to return to its caller unless the specified time plus the reference time is equal to or greater than LOBT. The equality case is normal. The greater than case is flagged as an error, but execution continues. SCR to return to it's caller.

[3]MDI LOBT should always match the spacecraft On-Board Time (OBT). See various parts of section 3 of EID-A for a description of OBT and LOBT.

4.2.5.3. THE INTERVAL COUNTER

SRC includes a counter that increments with each observing interval. The SIC instruction clears and starts the counter, while the GIC function returns its value to a register.

5. DEP COMMANDS

There are a number of telecommands that effect the operation of sequences. These are described in Table 6. These are all block telecommands using command Identifier 19_{16}. MDI340020 contains a complete description of all DEP telecommands.

NAME	FUNCTION	DATA WORDS
MBSQSTR	Start Sequence	Address
MBSQEND	Stop Sequence	None
MBSQREG	Set Register	Register,Value
MBSQDEVC	Set Device Default	Control [,Value]
MBSQMTMO	Load MTM Offsets	Offset1,Offset2
MBSQCONS	Configure on Shutter Close	None
MBSQCONR	Configure on Readout Done	None
MBSQMTLU	Load MTM Lookups	ID,Entry#,M1,M2 [,additional entries]

DEP Sequence Related Commands
Table 6

6. USAGE

This section describes the intention behind the implementation of SCR. It contains some examples and provides additional details about the interaction between SCR and the rest of the DEP software.

6.1. STARTING AND STOPPING SEQUENCES

Sequences are started by the MBSQSTR DEP command. Sequence processing is stopped by an MBSQEND command, an STS function, or an RET function from the highest sequence level, i.e., from a sequence started by telecommand. In all these cases, the DEP ceases sequence processing entirely. If a start sequence command is received while another sequence is running, all activity in the current sequence is terminated. All memory of prior context is lost; so, this is equivalent to an STS followed by a MBSQSTR, rather than a CLL.

6.2. LOOPS, BRANCHES, AND CALLS

There is no built-in looping capability in SCR, i.e. there is no FOR or WHILE. One repeats a group of functions using register based branch functions. Example 1 is a typical loop that one might use in a campaign. In this example, the part from labels A to B is the structure part of the campaign. That part could be in a Sequence Fragment and be replaced by a CLL.

```
        SIC                            Start the interval counter
        SET    R9, 120                 Set the repeat count
A:      SRT                            Set the reference time
        CON OC1 DM1
        TIM    -13.5
        TAP                            Take first of 10
        CON OC2 DM2                    Configure 2nd of 10
          ...
          ...
          ...
        CON    OCn DMn APn             Configure last of 10
        TIM    13.5
B:      TAP
        ......
        ......                         Do something for the other part of the interval
        ......
        GIC    R8                      Get the Interval counter
        BLT    &A R8 R9                Do this for [R9] intervals
```

Example 1
Looping in a Typical Campaign Sequence

6.3. INTERACTIONS WITH TELECOMMANDS

The DEP telecommand set includes commands to set registers. The concept behind this is to synchronize telecommand requests with the sequences. In particular, the sequences that service the magnetogram virtual channel interrupt normal structure observations. MVC services goes on for 5 - 10 minutes, at which time, the structure program resumes. One could realize this requirement with the fragment shown in Example 2. In this example, the sequence fragment SFST20 and SFMGRAM must include Set Reference Time (SRT) functions. The sequence SQMVC could contain a similar construct, i.e., branch on register set by telecommand, to determine when to return to this structure sequence.

```
        SID     XXXX XXXX          Sequence ID for struct program
A:      SET     R0 0               MVC flag
B:      SIC                        Start the interval counter
        SET     R9 120             Magnetogram every 2 hours
C:      CLL     SFST20             Call the 20 frame structure fragment
        BNE     &D R0 0            Has R0 been set
        GIC     R8
        BLT     &C R8 R9
        CLL     SFMGRAM            Call magnetogram fragment
        UBR     &B
D:      CLL     SQMVC              Call the MVC sequence
        UBR     &A                 Start over
```

Example 2
Synchronizing MVC Telecommand Interaction

6.4. CONFIGURING THE INSTRUMENT AND TAKING PICTURES

A typical sequence consists of a series of CON and TAP functions. A rigid, repeatable time sequence can be realized for most sequences by time tagging the TAP functions only. Example 3 shows part of a typical sequence. The CON specifies the position and rotation direction for the PAW and MTM's as well as the shutter beam path. Instrument reconfiguration requires a series of basic hardware functions. These are staged over the next several DEP time intervals following the CON function. The CON function need not be timed. SCR initiates optics reconfiguration at either shutter close time or upon completion of camera readout based on telecommand (See Table 6). Tables 7-1 and 7-2 show the sequence of events for the two cases. The tables show the automatic control of the Limb Tracker[4] as well as the activity of devices that are directly controlled by the CON function.

CON OC_n DM_n
TIM 1.5
TAP
CON OC_{n+1} DM_{n+1}
TIM 4.5
TAP

Example 3
Timing Between CON and TAP Functions

FUNCTION	TIME[5]	DEP HARDWARE EVENT	
		DEVICE	ACTIVITY
CON	R+1	MTM/PAW	SET POSITION
	R+2	SH	RECOCK
		CM	SEND HEADER 1
	C+2	PAW/MTM	MOVE (CW or CCW)
	R+3	SH	LOAD LOW BYTE
		LT	OPEN LOOP
		CM	LOAD HEADER DATA
	R+4	SH	LOAD HIGH BYTE
		LT	LOWER GAIN
		CM	LOAD INTEGRATE CMD
		CM	LOAD READOUT CMD
	P	LT	CLOSE LOOP
	P+1	LT	RESUME NORMAL GAIN
TAP	T+1	CM	SET TO AUTOMODE
		SH	ISSUE 3X or 1X CMD

DEP Hardware Configuration Event Timing
(Configure Flag Set To Readout Complete)
Table 7-1

[4]Limb Tracker control is asserted only if the PAW is moved.
[5]C = CON; R = Readout completed; P = PAW move complete; T=TAP

FUNCTION	TIME[6]	DEP HARDWARE EVENT	
		DEVICE	ACTIVITY
CON	S+1	PAW/MTM	SET POSITION
	S+2	PAW/MTM	MOVE (CW or CCW)
		LT	OPEN LOOP
	S+3	LT	LOWER GAIN
	R+2	SH	RECOCK
		CM	SEND HEADER 1
	R+3	SH	LOAD LOW BYTE
		CM	LOAD HEADER DATA
		SH	LOAD HIGH BYTE
	R+4	CM	LOAD INTEGRATE CMD
		CM	LOAD READOUT CMD
	P	LT	CLOSE LOOP
	P+1	LT	RESUME NORMAL GAIN
TAP	T+1	CM	SET TO AUTOMODE
		SH	ISSUE 3X or 1X CMD

DEP Hardware Configuration Event Timing
(Configure Flag Set To Shutter Closed)
Table 7-2

6.5. DIRECT DEVICE COMMANDS

The Direct Device Command Function allows the sequence to manipulate devices not normally operated from sequences. For example, one could step the alignment mechanism through some pattern or select different calibration wheel positions. In addition, one could write sequences that use DDC's rather than optics configuration tables. This approach may be useful in sequences such as lambda scans. Example 4 shows such a use. In the example, OCNULL is an Optics Configuration Block that has MTM and PAW direction bits set to NO MOTION; DMMEMx points to a DMA Control Block that place the output in some specific memory; and, IPSLICE points to an IP program that extracts a portion of an image. The example presumes that the same IP memories can be used for each image. In a real sequence, there may be some additional initialization steps.

[6] C = CON; S = Shutter Closed; R = Readout completed; P = PAW move complete; T=TAP

```
        SET   R0 45                      MTM initial Positions
        SET   R1 60
        SET   R2 120                     Initial time = 30
        SET   R3 0
A:      CON   OCNULL DMMEMx IPSLICE      Initial Config Blocks
        DDC   60 R0
        DDC   61 R1
        TIM   R2
        TAP                              Take the picture
        ADD   R1 3                       Set to next positions
        ADD   R0 6
        ADD   R2 48                      Set time 3 seconds ahead
        INC   R3
        BLT   &A R3 5                    Do for 5 frames
```

Example 4
Lambda Scan Using Direct Device Command

6.6. INTERACTIONS WITH THE IMAGE PROCESSOR

SCR has no facilities to allow direct sequence level interactions with in Image Processor. An Image Processor firmware program can be invoked indirectly by specifying the appropriate information in a configure function and the behavior of that program may, in some instances be influenced by the values in SCR registers.

6.7. TELEMETRY

Table 8-1 and 8-2 show housekeeping telemetry parameters that related to the sequencer status and operation. Document MDI340015 contains additional telemetry details.

MNEMONIC	DESCRIPTION
MKSHNEXP	Shutter Nominal Exposure
MKPADIR	PAW Default Direction
MKM1DIR	MTM1 Default Direction
MKM2DIR	MTM2 Default Direction
MKSHPATH	Shutter Default Beam Path
MKSQRG0	Sequence Register 0
MKSQRG1	Sequence Register 1
MKSQRG3	Sequence Register 3
MKSQRG4	Sequence Register 4
MKSQRG5	Sequence Register 5
MKSQRG6	Sequence Register 6
MKSQRG7	Sequence Register 7
MKSQRG8	Sequence Register 8
MKSQRG9	Sequence Register 9
MKSQRGA	Sequence Register A
MKSQRGB	Sequence Register B
MKSQRGC	Sequence Register C
MKSQRGD	Sequence Register D
MKSQRGE	Sequence Register E
MKSQRGF	Sequence Register F
MKSQID	Sequence Identifier
MKSQNUM	Sequence Number
MKSQFRM	Frame Number in Sequence
MKSQREF0	LOBT Reference 0
MKSQREF1	LOBT Reference 1
MKSQREF2	LOBT Reference 2
MKSQREF3	LOBT Reference 3
MKSQREF4	LOBT Reference 4
MKSQREF5	LOBT Reference 5

SCR Housekeeping Parameters
Table 8-1

MNEMONIC	DESCRIPTION
MKSQFQRT	Ref Time occurred in packet
MKSQFQRP	Frame Queue Pointer at Ref Time
MKSQFQPP	Frame Queue Pointer for Packet
MKSQFQNE	Number of Frame Queue Entries
MKSQRT1	Relative Time of Frame 1
MKSQSHO1	Shutter Open Time Frame 1
MKSQSHC1	Shutter Close Time Frame 1
MKSQPAA1	PAW Actual Position Frame 1
MKSQM1A1	MTM1 Actual Position Frame 1
MKSQM2A1	MTM2 Actual Position Frame 1
MKSQC1A1	CAL1 Actual Position Frame 1
MKSQC2A1	CAL2 Actual Position Frame 1
MKSQADR1	Sequence Address of Frame 1
MKSQRT2	Relative Time of Frame 2
MKSQSHO2	Shutter Open Time Frame 2
MKSQSHC2	Shutter Close Time Frame 2
MKSQPAA2	PAW Actual Position Frame 2
MKSQM1A2	MTM1 Actual Position Frame 2
MKSQM2A2	MTM2 Actual Position Frame 2
MKSQC1A2	CAL1 Actual Position Frame 2
MKSQC2A2	CAL2 Actual Position Frame 2
MKSQADR2	Sequence Address of Frame 2
MKSQRT3	Relative Time of Frame 3
MKSQSHO3	Shutter Open Time Frame 3
MKSQSHC3	Shutter Close Time Frame 3
MKSQPAA3	PAW Actual Position Frame 3
MKSQM1A3	MTM1 Actual Position Frame 3
MKSQM2A3	MTM3 Actual Position Frame 3
MKSQC1A3	CAL1 Actual Position Frame 3
MKSQC2A3	CAL3 Actual Position Frame 3
MKSQADR3	Sequence Address of Frame 3
MKSQRT4	Relative Time of Frame 4
MKSQSHO4	Shutter Open Time Frame 4
MKSQSHC4	Shutter Close Time Frame 4
MKSQPAA4	PAW Actual Position Frame 4
MKSQM1A4	MTM1 Actual Position Frame 4
MKSQM2A4	MTM4 Actual Position Frame 4
MKSQC1A4	CAL1 Actual Position Frame 4
MKSQC2A4	CAL4 Actual Position Frame 4
MKSQADR4	Sequence Address of Frame 4

SCR Housekeeping Frame Data
Table 8-2